CLAIMS

1. A method to increase a safety-integrity level of a Controller (10) for control of real-world objects, **characterized** by the steps of
- attaching to the said Controller (10) a safety-hardware unit (11) wherein the safety-hardware unit (11) communicates with the said Controller's CPU,
- downloading software with safety-related configuration data to the attached safety-hardware unit (11) and to the Controller (10),
- configuring the attached safety-hardware unit (11) to execute safety-function logic, which depends on the safety-related configuration data, and in an active or passive way set the Controller's (10) output values to a safe state for online safety control.

2. A method according to claim 1, **characterized** in that the Controller (10) has the capability of executing a set of non-safety-critical control functions, which set of non-safety-critical control functions is the same before as well as after the safety-hardware unit (11) is attached.

3. A method according to claim 2, **characterized** in that the configuring step comprises the additional steps of
- downloading to the attached safety-hardware unit (11) diagnostic information, which previously was automatically generated by a software tool as a result of user's configuration of the Controller (10) and which diagnostic information is used in the attached safety-hardware unit (11) during safety-critical control.

4. A method according to any previous claim, **characterized** in that access to a plurality of input and output values of a real-world object is obtained through a bus (14) connected between the Controller (10) and to an input/output unit (15) and the validity of the bus

(14) communication is verified in the attached safety-hardware unit (11).

5. A method according to any previous claim, **characterized** in that the timing supervision of the Controller (10) is verified in the attached safety-hardware unit (11).

6. A method according to any previous claim, **characterized** in that correct sequence of code logic is verified in the attached safety-hardware unit (11).

7. A method according to any previous claim, **characterized** in that correctness of memory content of the controller (10) is verified in the attached safety-hardware unit (11).

8. A method according to any previous claim, **characterized** in that a download of new control functionality logic to the Controller is verified in the attached safety-hardware unit (11).

9. A method according to any previous claim, **characterized** in that the attached safety-hardware unit (11) performs checks in order to allow only users logged on as safety-classified engineers and safety-classified operators to modify the control functionality logic and parameters.

10. A method according to claim 4, **characterized** in that the bus (14) communication verification logic in the attached safety-hardware unit (11) is implemented diverse.

11. A method according to claim 4, **characterized** in that the attached safety-hardware unit 11 is diverse generating a safety-related header for the bus (14) communication.

12. A method according to claim 11, **characterized** in that the Input/Output unit (15) has two diverse implementations, each verifying the correctness of the bus (14) traffic and each generating a safety-related header for the bus communication.

13. A method according to any previous claim, **characterized** in that the attached safety-hardware unit comprises a first and a second module in a redundant configuration, the second module is updated with data that exists in the first module at the time of a failure and the second module takes over the safety-related control of the control system from the first module if a failure of the first module is detected.

14. A method according to claim 13, **characterized** in that the a redundant Controller unit is attached to the Controller (10), which takes over in case of a failure of a primary Controller and the redundant Controller unit establishes communication with either the active first module or the active second module of the attached safety-hardware unit.

15. A Control System (20) intended for safety-related control of real-world objects, **characterized** in that it comprises

- a single main CPU handling the main processes of a Controller (10),

- an attached safety-hardware unit (11) comprising means to increase the safety-integrity level of the Controller and the comprising means to set the Controller's output values in a safe state for online safety control.

16. A Control System according to claim 15, **characterized** in that the Controller (10) has the capability of executing a set of non-safety-critical control functions, which set of non-safety-critical control functions is the same before as well as after the safety-hardware unit is attached.

17. A Control System according to claim 16, **characterized** in that it comprises
- means for downloading to the attached safety-hardware unit diagnostic information, which previously was automatically generated by a software tool as a result of user's configuration of the Controller and which diagnostic information is used in the attached safety-hardware unit during safety-critical control.

18. A Control System according to claim 17, **characterized** in that it comprises
- an input/output unit (15) connected to the Controller (10) by a bus and the validity of the bus (14) communication is verified in the attached safety-hardware unit.

19. A Control System according to claim 18, **characterized** in that the bus (14) communication verification logic in the attached safety-hardware unit (11) is implemented diverse.

20. A Control System according to claim 19, **characterized** in that the attached safety-hardware unit (11) is diverse generating a safety-related header for the bus (14) communication.